

Angular Context Buckets versus Binary LSH Contexts for Atlas Neural Networks

Motivation

Atlas neural networks benefit from context variables that partition the input space into regions with useful statistical properties. Traditional locality-sensitive hashing (LSH) produces binary contexts, while angular quantization of random projections produces ordered context variables with explicit neighborhood structure.

Binary LSH Context

- Compute multiple random projections.
- Quantize each projection by sign (+/-).
- The resulting binary vector indexes a context.

Advantages:

- Simple and computationally efficient.
- Good probabilistic locality: nearby inputs tend to share many bits.
- Well suited to high-dimensional partitioning.

Limitations:

- Context addresses have no natural ordering.
- Hamming distance only weakly represents geometric adjacency.
- Adjacent memory locations are generally unrelated geometrically.
- Sharing information between nearby contexts is not straightforward.

Angular Context

Choose two independent random projections:

- $x = u^T \cdot v$
- $y = w^T \cdot v$

Compute

$$\theta = \text{atan2}(y, x)$$

and quantize θ into angular buckets.

Properties:

- Angle is uniformly distributed for Gaussian distributed y and x .
- Nearby input directions produce nearby angles.
- Bucket adjacency is explicitly preserved.
- Buckets form a circular one-dimensional manifold.
- Small input perturbations usually remain within the same or neighboring buckets.

Unlike binary LSH, the bucket indices possess meaningful geometry.

Advantages for Atlas Networks

Angular contexts naturally support:

- Parameter sharing between neighboring buckets.
- Statistical smoothing across adjacent contexts.
- Hierarchical refinement by subdividing angular regions.
- Efficient interpolation between learned parameters.
- Adaptive resolution where additional model capacity is needed.

These operations are considerably less natural in Hamming space.

Multiple Angular Coordinates

Rather than computing a single angle, construct multiple independent projection pairs:

- (x_1, y_1)
- (x_2, y_2)
- ...
- (x_m, y_m)

Each pair contributes one circular context variable.

The resulting context representation consists of several ordered, low-dimensional variables instead of one large unordered binary code.

Geometric Interpretation

Different context mechanisms partition projection space differently:

- Decision trees: axis-aligned threshold regions.
- Binary LSH: hyperplane sign partitions.
- Angular quantization: wedge-shaped sectors with cyclic adjacency.

The wedge representation preserves local neighborhood relationships while remaining discrete.

Research Questions

- Does explicit bucket adjacency improve statistical efficiency?
- Can neighboring angular buckets safely share parameters?
- Is adaptive angular subdivision preferable to increasing binary code length?
- Can angular contexts reduce the amount of training data required?
- Are hybrid systems combining binary LSH and angular contexts superior to either representation alone?

Summary

Binary LSH preserves locality probabilistically but lacks explicit neighborhood structure. Angular quantization of random projection pairs produces discrete contexts with inherent geometric ordering. This additional structure may enable more efficient parameter sharing, smoother learning, and adaptive refinement within Atlas neural networks while retaining the computational simplicity of random projections.

Rather than ask only *how to partition the input space*, you can ask *what topology the context space itself should have*. Treating context variables as discrete objects with meaningful neighborhood relationships opens up algorithmic possibilities that may be absent from conventional binary LSH.